Planning and representation of joint human-agent space missions via constraint-based models

Planning and Representation of Joint Human-Agent Space Missions via Constraint-Based Models

Clauirton Siebra, Austin Tate and Natasha Lino

Artificial Intelligence Applications Institute
Centre for Intelligent Systems and their Applications
School of Informatics, The University of Edinburgh
Appleton Tower, Crichton Street, EH9 9LE, Edinburgh, UK
{c.siebra,a.tate,Natasha.Queiroz}@ed.ac.uk

Abstract. During space missions, the interaction between a spacecraft's planner and ground teams is very limited due to restrictions in communication. Commonly the role of such teams is reduced to receive the current state of a spacecraft and to send sequences of activities to it. However this concept of planning tends to change with the future advent of human interplanetary missions, so that astronauts become able to participate more actively in planning processes. This paper presents a planning approach, anchored in a constraint-based ontology, that considers the requirements of current space mission scenarios, but also provides the framework for future human missions. Furthermore we illustrate our ideas using a potential application in a Martian environment[1].

1. Introduction

The use of intelligent planning in long-term space missions has mainly been focused on providing levels of autonomy to spacecraft (e.g., orbiters) and, more recently, rovers (e.g., MER rovers). In fact, the use of a planner/scheduler (PS) in space is a very new experience if we consider that the first real application was in 1999, during the Remote Agent Experiment (RAX) [Muscettola *et al.*, 1998], on board of the *Deep Space One spacecraft*.

The purpose of RAX-PS [Jonsson *et al.*, 2000] was to generate plans that could be executed on board to achieve specified high-level goals. Its principal differences from classical STRIPS AI planning were that [Jonsson *et al.*, 2000]: actions can occur concurrently and can have different durations, and goals can include time and maintenance conditions.

The Automated Scheduling and Planning Environment (ASPEN) [Rabideau *et al.*, 1999] was intended to explore approaches complementary to RAX-PS. The principal focus was on classifying and repairing conflicts in the spacecraft models, which are described via the ASPEN Modelling Language (AML) [Smith *et al.*, 1998] .

The Continuous Activity Scheduling Planning Execution and Replanning (CASPER) system [Chien *et al.*, 2000] is an evolution of ASPEN that integrates repair planning with execution. The idea is to continuously replan around updated information coming from execution monitoring. CASPER was used in *Earth Observing-1* (EO-1) [Chien *et al.*, 2003], the first satellite in NASA's New Millennium Program Earth Observing series.

The beginning of rover missions to Mars created a new scenario for planning technology. This technology, however, is still based on earlier approaches. The OASIS system [Estlin *et al.*, 2003], for example, uses CASPER as a planner in its proposal of mixing techniques from both machine learning and planning to rover control.

Efforts, as in the OASIS system, aim to provide more autonomy to rovers. *Sojourner* (the first rover to operate on Mars), for example, travelled about 100 meters during its 90-day lifetime [Mishkin *et al.*, 1998]. However the Mars Exploration Rovers (*Spirit* and *Opportunity*) were designed to travel up to 100 meters per day. Autonomy is important because the rovers have intermittent and delayed communication with Earth. In fact the time of travel of a radio signal from Mars to Earth (about 10 minutes) precludes any real-time idea of continuous human operator control. Furthermore content issues of the *Deep Space Network* (DSN), an international network of antennas that supports interplanetary spacecraft missions, and planetary dynamics (position and rotation of the planets) also impose restrictions on communication.

According to [Ball *et al.*, 2001], the risk to human health during missions beyond Earth orbit (as exposure to high levels of radiation) is the greatest challenge to human exploration of deep space. Despite this threat, human interplanetary missions have seriously been discussed and some dates were already presented[2] as suitable to such missions.

The beginning of human interplanetary missions may lighten the communication delay problems. In this new

[1] This application was motivated by preliminary work to propose the application of I-X technology to a NASA experiment in the Mars Society' Desert Research Station in Utah, USA.

[2] In the XVI Congress of the Association of Space Explorers in 2000, for example, 06-May-2018 was presented as a suitable date to the first human mission to Mars due to the planets' orbits.

scenario, astronauts will carry out joint experiments with robots on planetary surfaces, so that several high-level goals and decisions could be taken into the work environment rather than made on Earth. This scenario will bring new requirements regarding joint human-agent planning [Allen and Fergunson, 2002; Sierhuis *et al.*, 2003], which differ to the current planning approaches to space missions.

This paper presents a framework that involves the use of shared models for task-directed cooperation between human and computer agents who are jointly exploring a range of alternative options for plans. We show that this framework can be used together with current planning technology, but that it also considers the required support to joint activities of human and agents.

The remainder of this document is structured as follows: section 2 discusses several planning requirements of human-agent space missions. Section 3 presents <I-N-C-A>, the constraint-based ontology that we are using to represent plans and processes. Section 4 details the application of <I-N-C-A> models to space missions and how users can manipulate such models via joint assistant agents (I-P^2). Section 5 demonstrates the use of this approach via a potential application in a Mars exploration scenario. Section 6 summarises a comparative discussion of related works, following by some conclusions and research directions.

2. Planning Requirements for Interplanetary Scenarios

This section splits the discussion about planning requirements for interplanetary missions into three different scenarios: the current scenario that looks for more autonomy for agents; missions with multiple software agents working collaboratively; and joint human-agent missions. This division corresponds to a natural evolution of space missions where robots (agents) can pave the way for human exploration.

2.1 Toward Autonomy

Recent space missions were carried out by individual agents (e.g., *Sojourner, Spirit* and *Opportunity*), which maintain non-continuous interaction with ground teams inside pre-defined communication windows. Despite the fact that *Spirit* and *Opportunity* were almost launched in the same period, they operated as individual missions on opposite sides of Mars (*Gusev Crater* and *Meridiani Planum* respectively) without any kind of collaborative activity.

The process currently used to control robots consists of uploading a sequence of "ground-created" commands to them. This process is very time-consuming and does not allow for dynamic adjustment of rover behaviour if anything unexpected happens, such as faults and new science opportunities. In this way, rather then consider planning as a batch process, planners such as CASPER are employing a continuous planning technique where they continually evaluate the current plan and modify it when necessary in accordance with the current state and resource information.

However in some situations an on-board library of Standard Operating Procedures (SOPs) can respond to unexpected events rather than the planner itself. A procedure or combination of procedures can, for example, quickly provide a safe state while the planner addresses the original plan fault. The *Sojourner* rover implemented a similar idea. According to [Mishkin *et al.*, 1998], if human error results in a sequence that ends with the rover being outside of communications range with its lander, for example, a procedure can be triggered so that the rover drives toward the lander.

SOPs present some interesting advantages. First they tend to be based on experiences, lessons learned or careful pre-design. Thus they are more robust. Second, they can be rapidly deployed. Third, a library of SOPs can be updated by ground teams or by the rover itself in a similar way to that described in [Estlin *et al.*, 2003]. Finally, the recurring tasks that rovers perform in different sites favour the use of SOPs. Unfortunately, as discussed in the planning literature, it is not possible to predict all situations in a real environment. Thus, automatic planning and pre-built SOPs must be used together.

An important kind of unexpected event is the detection of science events. The rover planner needs to decide if it will respond autonomously to such events, adding them to its agenda. If the rover is not prepared to deal with this new event, questions raised as a result of its analysis can be explicitly sent for additional analysis by ground teams.

2.2 Joint Multiagent Missions

According to [Clement and Barrett, 2003], there is a growing trend toward multi-spacecraft missions. About forty multi-spacecraft missions have been proposed, including formation flying teams and about 16 planned missions to Mars in the next decade. In this way, together with the requirements discussed in section 2.1, there is an additional need to support collaboration between the participants of such missions.

Collaboration between spacecrafts (rovers, landers, orbiters, etc.) will permit better use of time, multiple science instruments and communication channels. For that, the various spacecraft and devices must to be able to share knowledge and to plan considering the abilities of the whole group.

The teamwork theory [Cohen and Levesque, 1991] provides useful insights to the design of collaborative planning systems. The principal idea is that the team's joint activities do not consist merely of coordinated individual activities, but each participant needs, for example, to make commitments on reporting the status of their ongoing activities (progress, failure, or successful completion) and support the activities of others participants.

Several works [Levesque *et al.*, 1990; Grosz *et al.*, 1999; Kinny *et al.*, 1992; Jennings, 1994] have proposed frameworks using the teamwork concepts and, although they have different approaches to deal with different technical problems, they agree that agents involved in collaborative environments need to make commitments on joint activities, reach consensus on plans and also make commitments to the constituent activities of such plans.

2.3 Joint Human-Agent Missions

In this kind of scenario, humans will be complementary components to teams of robots. In the first instance, Mars ground/in-orbit teams can work as a front-end to Earth ground teams. Thus they can interact in real-time in the rover's execution if some unexpected event occurs. In this case, the team will only report to Earth issues that it is not able to deal with.

However in a more complex scenario, as described in [Sierhuis *et al.*, 2003], "Astronauts will live, work, and perform laboratory experiments in collaboration with robots inside and outside their spacecraft and habitats on planetary surfaces" (Figure 1). Thus, there is a need to provide a framework that supports a collaborative mixed-initiative style of planning [Tate, 1997], which fills the gap between the abilities of automated reasoners and the needs of human decision makers [Allen and Fergunson, 2002].

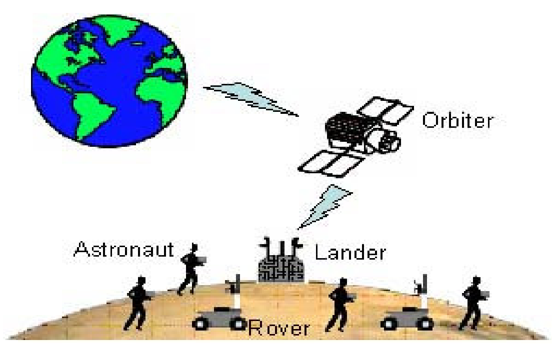

Figure 1: Potential scenario for future Mars exploration missions

3. <I-N-C-A> Ontology

<I-N-C-A> (Issues - Nodes - Constraints - Annotations) [Tate, 2003] is a general-purpose ontology that can be used to represent synthesised artefacts, such as plans, in the form of a set of constraints on the space of all possible artefacts in the application domain. The use of <I-N-C-A>, in this work, aims to underpin the representation of collaborative activities, respecting the requirements discussed in the section 2. The next subsections describe each of the <I-N-C-A> components.

3.1 Issues

Issues in the <I-N-C-A> representation may state the outstanding questions to be handled and can represent unsatisfied objectives, questions raised as a result of analysis, etc. New issues can imply potential further nodes as constraints to be added.

We have adopted the gIBIS [Conklin and Beneman, 1988] orientation of expressing these issues as any of a number of specific types of question to be considered [Selvin, 1999]. The types of questions advocated[3] are:

1. Deontic questions: What should we do?
2. Instrumental questions: How should we do it?
3. Criterial questions: What are the criteria?
4. Meaning or conceptual questions: What does X mean?
5. Factual questions: What is X? Is X true?
6. Background questions: What is the background to this project?
7. Stakeholder questions: Who are the stakeholders of this project?
8. Miscellaneous questions: to act as a catch all.

The first 5 of these are likely to be the most common in our task support environment. This is similar to the *Question-Option-Criteria* approach [MacLean, 1991] (itself used for rationale capture for plans schema libraries in our earlier work [Polyak and Tate, 1998]), and to the mapping approach used in Compendium (a Computer-Supported Cooperative Working tool) [Selvin *et al.*, 2001].

3.2 Nodes

Nodes describe components that are to be included in the artefact. Nodes can themselves be artefacts that can have their own structure with sub-nodes and other <I-N-C-A> described refinements associated with them.

When <I-N-C-A> is being used to describe plans as processes, the nodes are usually the individual activities or their sub-activities. They are usually characterized by a "pattern" composed of an initial verb followed by any number of parameter objects, noun phrases, and qualifiers or filler words describing the activity. For example:

(**transport** *rock* from *crater* to *lander*)

3.3 Constraints

Constraints restrict the relationships between the nodes to describe only those artefacts within the artefact space that meet the requirements. The constraints may be split into "critical constraints" and "auxiliary constraints" depending on whether some constraint managers (solvers) can return them as "maybe" answers to indicate that the constraint being added to the model is okay so long as other critical constraints are imposed by other constraint managers. The *maybe* answer is expressed as a disjunction of conjunctions on such critical or shared constraints. The "yes/no/maybe"

[3] Based on work by J.Conklin (personal communication) who extended the empirically derived question types in [Selvin, 1999].

constraint management approach is detailed in [Tate, 1995].

The choice of which constraints are considered critical and which are considered auxiliary is itself a decision for an I-X application. Specific decisions on how to split the management of constraints within such an application are required. For example, a temporal activity-based planner would normally have object/variable constraints (equality and inequality of objects) and some temporal constraints (maybe just the simple “before” constraint: {*before time-point-1 time-point-2*}) as the critical constraints. But, in a 3D design or a configuration application object/variable and some other critical constraints (possibly spatial constraints) might be chosen. It depends on the nature of what is communicated between constraint managers in the application.

3.4 Annotations

Annotations add complementary human-centric and rationale information to the plan, and can be seen as notes on their components.

3.5 Plans

Each plan is made up of a set of “Issues”, “Nodes”, and a set of “Constraints” which relate those nodes and objects in the application domain. Annotations can be added to the overall plan as well as specifically on any component of the plan. Figure 2 shows part of the <I-N-C-A> specification for plans.

```
plan = element plan
{
      element plan-variable-declarations
            { element list { plan-variable-declaration* } }?
      & element plan-issues
            { element list { plan-issue* } }?
      & element plan-issue-refinements
            { element list { plan-issue-refinement* } }?
      & element plan-nodes
            { element list { plan-node* } }?
      & element plan-refinements
          { element list { plan-refinement* } }?
      & element constraints
            { element list { constrainer* } }?
      & element annotations
            { map }?
}
```

Figure 2: Part of the <I-N-C-A> schema for plan specifications

4. Employing <I-N-C-A> Models

We “deliver” useful functionality based on the <I-N-C-A> ontology via I-X Process Panels (I-P^2) [Tate *et al.*, 2002]. A panel shows the current state of collaborative planning (from the perspective of the panel’s user) through the presentation of the current items of each of the four sets of entities comprising the <I-N-C-A> model. I-P^2 has been demonstrated in several different scenarios such as Coalition and Multinational Forces Command and Control [Allsopp *et al.*, 2002] and Search and Rescue Coordination [Siebra and Tate, 2003]. Here we discuss how it also could be applied to deal with the requirements listed in the section 2.

4.1 The Collaborative Framework

The principal objective of I-P^2 is to provide support to the joint planning and execution activities of a team. In a Mars exploration mission, for example, such a team is made of astronauts, rover, orbiters, etc. The *I-Space* tool (Figure 3) is an I-P^2 resource that manages the relationships of a specific agent (e.g., Astronaut-3) to others agents and external services (e.g., Orbiter-1 is a service that provides a communication interface, radiation measurement and position tracking). Particular actions can be associated to each of the relations (e.g., “delegate to” action is possible only to subordinate agents.

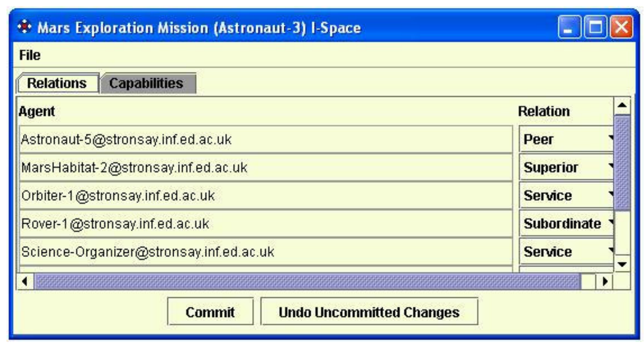

Figure 3: View of I-Space tool to the Astrounat-3

The capabilities of these agents will be used together to perform the high-level activity of the mission, which we describe as “Explore Mars”. During the planning process such an activity can be decomposed into several sub-activities, forming a hierarchical tree, as exemplified in Figure 4. This tree is dynamic so that unexpected events can add, update or delete activities.

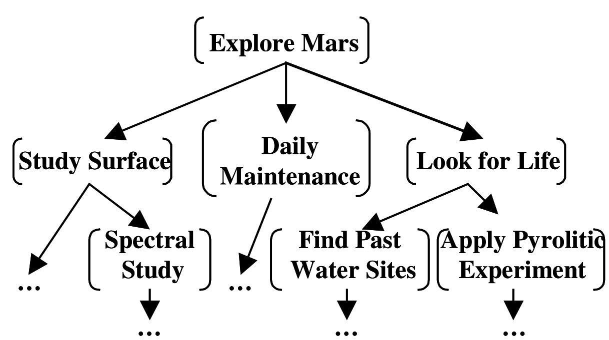

Figure 4: Example of hierarchical decomposition of activities

Agents receive their activities via an I-X Process Panel (Figure 5), whose contents, along with the current context and state of the collaboration, are used to generate dynamically the support options the tool provides. For example, associated with a particular activity node might be suggestions for performing it using known Standard Operating Procedures, for invoking an agent offering a service of some kind, or for delegating the activity to some other agent in the team.

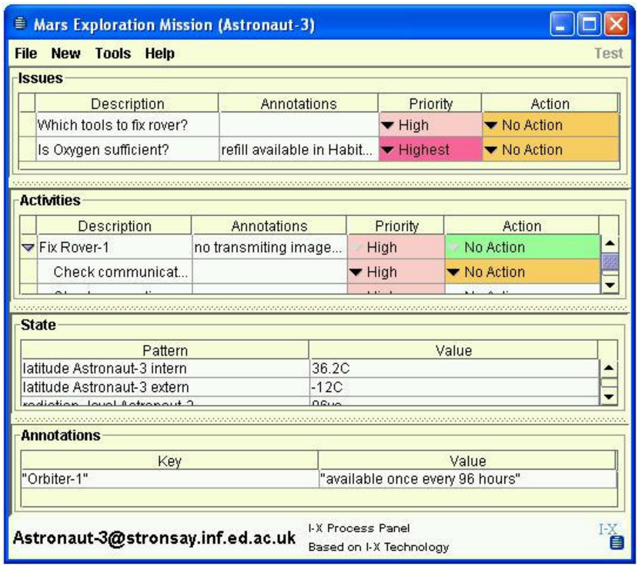

Figure 5: I-X Process Panel and its 4 sub-panels

For any activity on the panel, an "Action" column shows its current execution status and the available options to perform the activity. Colours indicate the readiness of the item for current execution:

- White indicates that the item is not currently ready for execution (i.e., some temporal ordering, precondition or other constraints are not yet met);
- Orange indicates that the action is ready to perform and that all preconditions and constraints are met;
- Green indicates that the item is currently being performed;
- Blue indicates successful completion;
- Red indicates a failure for which failure recovery planning steps might be initiated.

The set of "Actions" available to perform any item on the panel is available through a menu. This is dynamically generated and context-sensitive, reflecting the knowledge of the capabilities of the current agent, other team members through their panels and the various services available. This includes the selection from any known plans or Standard Operating Procedures that match the item. I-P^2 has a domain editor that allows the creation, maintenance and, ultimately, the publication of such Standard Operating Procedures.

Activities and other panel items can be passed from one panel to another (or to capable services or other agents). These can pass back "progress" and "completion" (success/fail) reports to the original sender of the item. This provides a way to monitor activity progress, receive back milestone reports, and check off the completion of activities. Incoming completion reports and information about the current state sent as constraints can trigger later activities to be executable as temporal or other constraints are satisfied.

In this way, by having a clear description of the different components within a synthesised plan (as activities and constraints), <I-N-C-A> allows such plans (or part of them) to be manipulated and used separately from the environment in which they are generated. This feature enables agents to individually work on different parts of the plan, but without losing the awareness of collaboration.

4.2 I-P^2 for Astronauts on the Move

Currently I-P^2 has been developed to run on desktop-based computers. However in several scenarios, such as interplanetary surface exploration, where human users will explore scenarios by themselves, the I-P^2 technology needs to be adapted to run in limited (PDA-like[4]) devices. Such platforms present several limitations as in processing power, memory, screen space, connection bandwidth, etc.

Considering these aspects, planning applications aimed at desktop platforms cannot be straightforwardly adapted and applied to limited devices. There is a need to investigate new approaches to develop planning mechanisms that respect the existing limitations.

An initial effort [Lino *et al.*, 2003] of our project is investigating this adaptation thread. The first focus is on how to show <I-N-C-A> planning information on limited devices. The ongoing visualisation framework is been based on scenario characteristics, such as the agent that is requesting planning information and its preferences, the planning information being requested, the device where the planning information will be delivered and its capabilities, and the available resources (map or sketch tools, GPS, etc.). Using such characteristics, an intelligent mapping process could deliver planning information to several kinds of devices, looking for the most "legible" and appropriate way to present it.

4.3 I-P^2 for "Rovers"

I-P^2 normally provides a graphical interface to an I-X agent. Software agents (e.g., rovers, landers and orbiters) could make use of the underlying I-X collaborative framework implementing these agents. An example is shown in Figure 6 for software agents based on the BDI architecture [Rao and Georgeff, 1995]. According to the figure, the software agents have the same function as users.

---

[4] PDA-like devices present similar technical features of PDA's (Personal Digital Assistants), but possibly are adapted to the environment that they will be used in.

First they must believe in the state of the world represented by the model. Second they must desire to hold the joint objectives. Finally they must adopt a plan and intend to perform its set of activities.

I-X agents support their integration with other systems via an open plug-in interface, which allows the definition of handlers to manipulate the <I-N-C-A> models. In the same manner, this interface also enables the use of external services, such as the CASPER planner, each potentially using a different internal logic and working with a different scope. The key advantage of an open interface is that the final system can be based on technologies already available.

On the other hand, a heterogeneous approach makes the framework difficult to validate in respect of all the models and procedures, and to ensure that they do not conflict. The IDEA (Intelligent Distributed Execution Architecture) framework [Muscettola *et al.*, 2003], for example, discusses the advantages of unifying planning and execution modules. Future development of I-X aims to provide more assistance via its planner (I-Plan), so that users have the option of using a unique technology, or representation and problem solving capability of distinct services.

5. Using I-P^2 during an One-Sol Mission

This section exemplifies the features of I-P^2 via a fictitious one-sol (one Martian day) mission. The duration of daylight is itself a constraint to the mission, because some rovers have devices that only work with solar energy (e.g., APXS spectrometer). The mission is realised during the summer in the Southern hemisphere on Mars because of higher temperatures. However this temperature may change significantly due to perihelic dust-storm activity, requiring continual re-planning.

The mission has several decision-making levels. The ground team on Earth sets the macro goals, sharing the activities with the *Mars-Habitats*. Each *Mars-Habitat* has one or more *exploration teams*, which are composed of a lander, two rovers (r_1 and r_2) and two astronauts (a_1 and a_2). Orbiters provide some auxiliary services. In our example there are two principal objectives for each exploration team: studying the surface of Mars (activity assigned to rovers) and looking for some sign of life (activity assigned to astronauts). The lander provides a higher bandwidth communication channel (e.g., for high-quality images transmission) and a mobile micro-laboratory.

Supposing an exploration team has a sol-mission set to start at 06:00h and to finish at 18:00h. During this period, information on the current state of the environment can be passed to I-X Process Panels via "world state" constraints. These might come from sensors directly, or from some analysis or reporting system. Additional I-X viewers (Figure 7 and 8) can display the world state in a more natural way to astronauts, and also work as a data input mechanism.

At 06:00h the team members begin their work together with their panels, which represent what they need to do (activities), directions of how to do these (sub-activities), and what cannot be done (constraints). Issues will appear during the performance of the activities. For example, at 10:00h the temperature quickly starts to decrease, so that the exploration activities of a_1 and a_2 are paused. This fact also generates a new issue: "What should we do?". Thus the I-X agent, using its own capability or external services, returns possible actions to deal with this new issue. Actions represent new sets of nodes that can be added to the model, respecting the constraints currently active. When an action is selected, its constraints are propagated so that they restrict the creation of new nodes.

At 12:00h the *Mars-Habitat* notices a failure in rover r_2. Then it sets a new activity with *highest* priority to astronaut a_1: "Fix r_2". For that, a_1 makes use of the library of plans, which contains standard operating procedures about how to discover and fix rover problems. This new activity delays a_1, however *Mars-Habitat* knows this fact because a_1, via its I-P^2, is always sending reports about its performance. In this way, *Mars-Habitat* is able to predict global failures and re-plan new schedules.

At 14:30h r_1 finds a possible sign of life during its soil analysis, whose investigation is not part of its activities. However the panels have a shared model of the world so that when this science event is added in the model by r_1 as

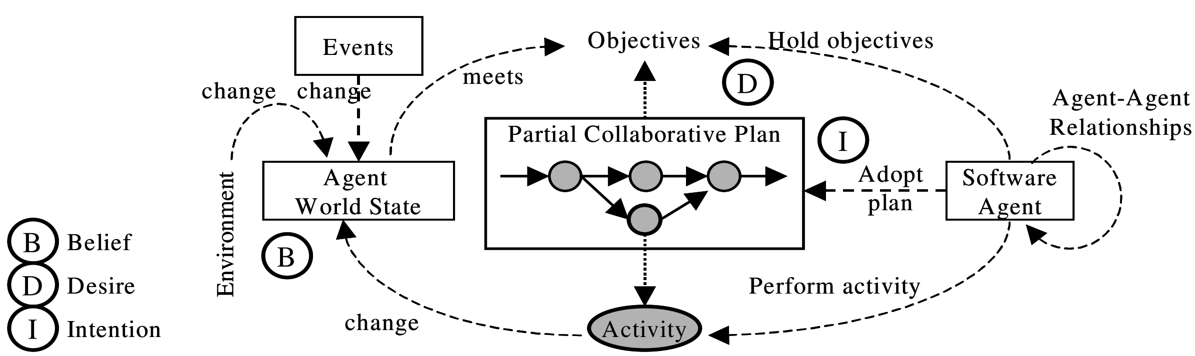

Fig. 6: Planning process overview for I-X software agents

a “world state” constraint, a_1 and a_2 are able to recognise the event. In this case, the location of the event determines which panel will create a new activity (“examine event”), in accord with the area assigned to the astronauts.

At 16:05h the orbiter reports a strong solar activity, which increases the emission of solar energetic particles. Thus all outdoor human activities are cancelled and a_1 and a_2 are not able to complete all their activities. In this case the *Mars habitat* team, supported by its I-P^2, must provide a new plan that involves the remaining activities of the day. If this plan is not possible, the team reports the situation to the Earth team.

Figure 7: I-X Map Viewer. This example uses a JPG image as the surface, however the viewer enables the plug in of a PDS (Planetary Data System) layer to manipulate real surface data.

Figure 8: In the I-X 3D Viewer, objects are modelled via VRML and I-P^2 imports them using the Java 3D API

6. Related Works

We can outline a brief parallel between the I-X approach and some multiagent systems that support joint human planning activities. In CODA (Coordination of Distributed Activities) [Myers *et al.*, 2001], for example, each user declares the kinds of plan changes (*Plan Awareness Requirements* - PAR) that are of interest to him/her. As users develop plans using a structured plan editor, their activities can be monitored so that changes that match declared interests are automatically forwarded to the person who declared interest in them.

In a different way, the DSIPE distributed planning framework [DesJardins and Wolverton, 1999] generates PARs through analysis of causal plan structures rather than authored by humans. Each DSIPE agent has a complete model of its own subplan and a partial model of subplans being developed by other agents. This partial model is represented by nodes that are not expanded and serve as placeholders for attaching PARs.

In I-X agents (I-P^2), the idea of PARs can be expressed by constraints that indicate, for example, the necessary conditions to perform an activity. I-X agents provide mechanisms to send such constraints (together with activities and plans) to other agents. However this planning information is normally only shared between the activity sender (in general a superior agent) and the activity receiver (in general a subordinate agent).

We can note that, differently of DSIPE, I-X agents do not have a partial model of the other agents’ plans. In fact the DSIPE approach is useful to rapidly detect conflicts before the planning merge process. However this approach requires an extra reasoning to reduce the communication requirements while maintain the consistency of all local plan models.

In I-X we are studying alternatives to this approach. An option is to leave the problem of conflict detection to agents that account for merging subplans. As agents are planning in a continuous way, the merging agent is always receiving new planning information so that it is also able to rapidly detect and resolve such conflicts.

7. Conclusion and Future Directions

The framework presented in this paper aims to provide a collaborative environment, which can be used during current and future interplanetary missions. Firstly, the open style of architecture enables the use/integration of our approach with current space-related technologies. Earlier examples of our work have been used for a variety of ground-based telecommand (e.g., EUMETSAT), assembly, integration and test (e.g., for Ariane IV), and ground segment scheduling (e.g., UK SkyNet) applications.

Secondly, the framework provides the same task support to agents, independently of whether they are software or humans. Similar applications in multiple unmanned aerial vehicle (UAV) mixed-initiative mission planning are also under investigation. Incompatibility

Thirdly, the use of standard operating procedures is supported by a domain editor tool, which enables the editing and publication of procedures ever at runtime.

Finally, the set of I-X Process Panels implements concepts of teamwork theories, enabling an effective collaboration between the participants. Future directions of this work aims to extend the I-X planner (I-Plan) abilities so that users have the option of using the framework as a unique and integrated system, rather than using additional reasoners.

Acknowledgement

Scholarships for Clauirton Siebra and Natasha Lino are sponsored by CAPES Foundation under Processes No. BEX1944/00-2 and No. BEX2092/00-0. This material is based on research within the I-X project, sponsored by the Defense Advanced Research Projects Agency (DARPA), US Air Force Research Laboratory under agreement number F30602-03-2-0014, the EPSRC-Sponsored UK Advanced Knowledge Technologies Project, and others sources.

References

J. Allen and G. Ferguson. Human-Machine Collaborative Planning. In *Proc. 3rd International NASA Workshop on Planning and Scheduling for Space*. Houston, Texas, USA, 2002.

D. Allsopp, P. Beautement, J. Bradshaw, E. Durfee, M. Kirton, C. Knoblock, N. Suri, A. Tate, and C. Thompson. Coalition Agents Experiment: Multiagent Cooperation in International Coalitions, *IEEE Intelligent Systems, 17*(3):26-35, 2002.

H. Ball, C. Evans, J. Ballard and J. Ball. *Safe Passage: Astronaut Care for Exploration Missions*. National Academy Press, 2001.

S. Chien, R. Knight, A. Stechert, R. Sherwood and G. Radibeau. Using Iterative Repair to Improve Responsiveness of Planning and Scheduling. In *Proc. 5th International Conference on AI Planning and Scheduling*. Breckenridje, CO, USA, 2000.

S. Chien, *et al.* Autonomous Science on the EO-1 Mission. In *Proc. International Symposium on AI, Robotics and Automation in Space*. Nara, Japan, 2003.

B. Clement and A. Barrett. Continual Coordination through Shared Activities. *In Proc. 2nd International Conference on Autonomous and Multi-agent System*. Melbourne, Australia, 2003.

P. Cohen and H. Levesque. Teamwork, *Special Issue on Cognitive Science and Artificial Intelligence*, 25, 487-512, 1991.

J. Conklin and M. Begeman. gIBIS: A hypertext tool for exploratory policy discussion. *ACM Transactions on Office Information System,* 4(6), 303-331, 1988.

M. DesJardins and M. Wolverton. Coordinating a Distributed Planning System. *AI Magazine*, 20(4):45-53, 1999.

T. Estlin, R. Castano, R. Anderson, D. Gaines, F. Fisher and M. Judd. Learning and Planning for Mars Rover Science. *In Proc. IJCAI Workshop on Issues in Designing Physical Agents for Dynamic Real Time Environments: World Modelling, Planning, Learning and Communicating*. Acapulco, Mexico, 2003.

B. Grosz, L. Hunsberger and S. Kraus. Planning and Acting Together, *AI Magazine, 20*(4), 1999, 23-34.

N. Jennings. Commitments and conventions: the foundation of coordination in multiagent systems, *The Knowledge Engineering Review*, 8, 1994.

A. Jonsson, P. Morris, N. Muscettola and K. Rajan. Planning in Interplanetary Space: Theory and Practice. *In Proc. 5th International Conference on AI Planning and Scheduling*. Breckenridje, CO, USA, 2000.

D. Kinny, M. Ljungberg, A. Rao, G. Tidhar and E. Werner. Planned Team Activity, *Proc. 4th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, Rome, Italy, 1992.

W. Kunz and H. Rittel. Issues as Elements of Information Systems, Working Paper no. 131, Institute of Urban and Regional Development, University of California, Berkeley, CA, 1970.

J. Levesque, P. Cohen and J. Nunes. On Acting Together, *Proc. AAAI National Conference on Artificial Intelligence*, Menlo Park, CA, 1990.

N. Lino, A. Tate, C. Siebra and Y. Chen-Burger. Delivering Intelligent Planning Information to Mobile Device Users in Collaborative Environments. In *Proc. IJCAI Workshop on AI , Information Access and Mobile Computing*. Acapulco, Mexico, 2003.

A. MacLean, R. Young, V. Berllotti and T. Moran. Questions, Options, and Criteria: Elements of Design Space Analysis. *Human-Computer Interaction*, 6(3-4):201-250, 1991.

A. Mishkin, J. Morrison, T. Nguyen, H. Stone, B. Cooper and B. Wilcox. Experiences with Operations and Autonomy of the Mars Pathfinder Microrover. In *Proc. IEEE Aerospace Conference*. Aspen, CO, USA, 1998.

N. Muscettola, P. Nayak, B. Pell and B. William. Remote Agent: to boldly go where no AI System has gone before. *Artificial Intelligence* 103(1-2):5-48, 1998.

N. Muscettola, G. Dorais, C. Fry, R. Levinson and C. Plaunt. IDEA: Planning at the Core of Autonomous

Reactive Agents. In *Proc. International Symposium on AI, Robotics and Automation in Space*. Nara, Japan, 2003.

K. Myers, P. Jarvis and T. Lee. CODA: Coordinating Human Planners. *Proceedings of the European conference on Planning*, Toledo, Spain, 2001.

S. Polyak and A. Tate. Rationale in Planning: Causality, Dependencies and Decisions. *Knowledge Engineering Review*, 13(3):247-262, 1998.

G. Rabideau, R. Knight, S. Chien, A. Fukunaga and A. Govindjee. Iterative Repair Planning for Spacecraft Operations Using The ASPEN System. In *Proc. International Symposium on AI, Robotics and Automation in Space*. Noordwijk, Netherlands, 1999.

A. Rao. and M. Georgeff. BDI Agents: From Theory to Practice. In *Proc. 1st International Conference on Multi-Agent Systems*. San Francisco, CA, USA, 1995.

A. Selvin and S. Shum. Rapid Knowledge Construction: A Case Study in Corporate Contingency Planning Using Collaborative Hypermedia. *Knowledge and Process Management*, 9(2):119-128, 2002.

C. Siebra and A. Tate. I-Rescue: A Coalition Based System to Support Disaster Relief Operations. In *Proc. 3rd International Conference on Artificial Intelligence and Applications*. Benalmadena, Spain, 2003.

M. Sierhuis, J. Bradshaw, A. Acquisti, R. Hoof, R. Jeffers and A. Uszok. Human-Agent Teamwork and Adjustable Autonomy in Practice. In *Proc. International Symposium on AI, Robotics and Automation in Space*. Nara, Japan, 2003.

B. Smith, R. Sherwood, A. Govindjee, D. Yan, G. Rabideau, S. Chien and A. Fukunaga. Representing Spacecraft Mission Planning Knowledge in ASPEN. In *Proc. AI Planning Systems Workshop on Knowledge Acquisition*, Pittsburgh, PA, 1998.

A. Tate. Integrating Constraint Management into an AI Planner. *Journal of Artificial Intelligence in Engineering*, 9(3):221-228, 1995.

A. Tate. Mixed-Initiative Interaction in O-Plan. In *Proc. AAAI Spring Symposium on Computational Models for Mixed Initiative Interaction*. Stanford, CA, USA, 1997.

A. Tate, J. Dalton and J. Stader. IP^2 – Intelligent Process Panels to Support Coalition Operations, *Proc. 2nd International Conference on Knowledge Systems for Coalition Operations*, Toulouse, France, 2002.

A. Tate. <I-N-C-A>: an Ontology for Mixed-Initiative Synthesis Tasks. In *Proc. IJCAI Workshop on Mixed-Initiative Intelligent Systems*, Acapulco, Mexico, 2003.